UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/824,450	04/14/2004	Adam H. Leventhal	03226.417001;SUN040872	6601

32615 7590 11/15/2007

OSHA LIANG L.L.P./SUN
1221 MCKINNEY, SUITE 2800
HOUSTON, TX 77010

EXAMINER	
LEE, MARINA	

ART UNIT	PAPER NUMBER
2192	

NOTIFICATION DATE	DELIVERY MODE
11/15/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lord@oshaliang.com
hernandez@oshaliang.com
DOCKETING@OSHALIANG.COM

Office Action Summary	Application No.	Applicant(s)	
	10/824,450	LEVENTHAL ET AL.	
	Examiner	Art Unit	
	Marina Lee	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 April 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-13* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *April 14, 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed April 14, 2004.

2. Claims 1-13 are pending and have been examined.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and In *re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969). A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. As to examined claims **1-13 (of the current application – 10/824450**) are provisionally rejected on the ground of nonstatutory double patenting over referenced claims (**1-8,11-17, and 19-20**) (of the co-pending application – **10/713411** – annotated date: 05/15/2007) and the referenced claims **1-16** (of the co-pending application – **10/713,409** – annotated date:05/08/2007). This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

The conflict claims: **1-13** of (the instant application) , **1-8, 11-17, and 19-20** of the referenced co-pending application (**10/713,411**) and, 1-16 of the referenced co-pending application (**10/713,409**) are not identical; however, by broadly interpretation, claim 1 & 8 of the instant application, are merely teaches trace an instrumented program on a x86 processor where as claims 1 &13 and

claims: 1&10 of the referenced co-pending application(10/713,411 and 10/713,409 respectively) cover tracing an instrumented program (without specifying any particular processor architecture).

The conflict claims: 1&8 of the examined application claim are not patentably distinct from the reference claims (1 &13) and (1& 10) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s) combine with teaching by Tamches, "Fine-Grained Dynamic Instrumentation of Commodity Operating System Kernels), which emphasis on Splicing on Architecture having variable length instruction (e.g., x86 – see 4.6.1, page 66-67).

Therefore, the examined claims 1 and 8 are anticipated by the referenced co-pending claims 1 &13 and 1&10 of application10/713,411 and 10/713,409 respectively.

Hence, the examined claims 1-13 and referenced co-pending claims: 1-8,11-17, and 19-20 (of 10/717,409) and 1-16 of (10/717,409) are not patentably distinct and as such are also unpatentable for obvious-type double patenting. *See, e.g.,In re Berg, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); In re Goodman, 11F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); and In re Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985). See also MPEP§ 804*

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-4 and 6-7 are rejected under 35U.S.C. 102(e) as being anticipated by Tamches, "Fine-Grained Dynamic Instrumentation of Commodity Operating System Kernels", University of Wisconsin, 2001.

As to claim 1, Tamches discloses, a method of tracing an instrumented program on a processor having an x86 architecture, comprising:

triggering a probe in the instrumented program (*see –e.g., "probe point" of IBM's Dynamic probe code (Dprobes), page 15, "instrumentation point" of code patch, page 49, and "entry point" of the original code to unconditional branch, page 62*);

obtaining an original instruction associated with the probe (see "instrumentation point" of code patch ,page 49, and "entry point" of the original code to unconditional branch, page 62) ;

loading the original instruction into a scratch space (*e.g., allocate the code patch and data heaps are load into kernel address space using driver /dev/kernist; and springboard (scratch space) – see KernInst Architecture 3.1, Page 26-27 and Fig. 4.9, and page 62*);

loading a jump instruction for the x86 architecture into the scratch space wherein the jump instruction includes a next program counter value (*e.g., overwrite a single branch or jump instruction x86 by writing a one-byte trap or .. .and jump to the appropriate code patch – see page 66-67, section 4.6.1 and code splicing overwrite instruction of Figure 4.9, page 62*);

executing the original instruction in the scratch space using a thread (*e.g., the code patch is executed at boot time or run-time of the kernel and the springboard execution – page 49, page 62, and associated text*); and

executing the jump instruction in the scratch space using the thread (*e.g., the code patch is executed at boot time or run-time of the kernel and the springboard execution – page 49, page 62, and associated text*).

As to claim 2, Tamches discloses further comprising:

emulating the original instruction to determine a program counter value if the original instruction is a control-flow instruction; and returning control to the thread at an address of the program counter value if the original returning control to the thread at an address of the program counter value if the original instruction is control-flow instruction (e.g., the relocating the overwritten instruction to the code patch semantic section 4.2.1, page 51-52).

As to claim 3, Tamches discloses further comprising:

determining the next program counter value by incrementing a value of a program counter using a size of the original instruction (e.g., increment the program counter for each original function call or event counter increment – *see Figure 5.1, page 72 and associated text*).

As to claim 4, Tamches discloses wherein the probe corresponds to a trap (e.g., trap handle use in splicing and trap instruction – see page 15-17 and section 4.6.1, page 66-67).

As to claim 6, Tamches discloses wherein the scratch space is allocated on a per-thread basis (e.g., event interval accumulation per thread – see page 77 and related text).

As to claim 7, wherein the instrumented program is executed on a multi-thread architecture (e.g., even interval accumulation per multiple thread – see page 66 and related text).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 5 and 8-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tamches, "Fine-Grained Dynamic Instrumentation of Commodity Operating System Kernels", University of Wisconsin, 2001 and in view of Mattson, Jr. et al., (hereinafter – Mattson), (U.S. Patent No. 6,327,704 B1).

As to claim 5, it is noted that Tamches does not explicitly disclose wherein obtaining the original instruction comprises: searching a look-up table using a program counter value, wherein the look-up table comprises the original instructions associated with the probe and an address associated with the original instruction. However, Mattson, in an analogous art, teaches dynamic translator using branches of the multi-branch-jump instruction are executed, the

dynamic backpatching code enables a backpatcher that replaces the corresponding entry in the translated multiple branch-jump table using with pointers to the address of the translated target address, e.g., direct jump backpatch 1010 determines, using any of a variety of know techniques, such as search and compare techniques, whether such ordinal target address has been enter by memory manager 720 in translated instruction look-up table 232 (see Mattson, Abstract, col. 12: 1-51, Look-up table 232, col. 27: 31-42, and related text).

It would have been obvious to a person having ordinary skill in the art at the time of invention was made to use dynamic backpatching determination through look-up table 232 of Mattson in dynamic instrumentation (e.g., handle to look up address) of Tamches for increasing the efficiencies of the dynamic translation as once accomplished by Mattson (see Mattson, col. 1: 48-63).

As to claim 8, Tamches discloses a system for tracing an instrumented program on a processor having an x86 architecture, comprising:

a thread configured to execute the instrumented program (*e.g., allocate the code patch and data heaps are load into kernel address space using driver /dev/kernist; and springboard (scratch space) – see KernInst Architecture 3.1, Page 26,-27 and Fig. 4.9, and page 62*);

a scratch space arranged to store the original instruction and the jump instruction (*e.g., kernel address space and springboard (available space) – see KernInst Architecture 3.1, Page 26,-27 and Fig. 4.9, and page 62*) ; and

an execution facility for executing the original instruction to collect data and executing the jump instruction, wherein the execution facility is a processor based on the x86 architecture (*e.g., overwrite a single branch or jump instruction x86 by writing a one-byte trap or .. .and jump to the appropriate code patch – see page 66, section 4.6.1 and code splicing overwrite instruction of Figure 4.9, page 62).*

It is noted that Tamches does not explicitly disclose a look-up table arranged to store an address and a corresponding original instructions; a trap handler configured to halt execution of the thread when a trap instruction is encountered, use an address of the trap instruction to obtain the corresponding original instruction from the look-up table, and generated a jump instruction to an address in the instrumented program trap. . However, Mattson, in an analogous art, teaches dynamic translator using branches of the multi-branch-jump instruction are executed, the dynamic backpatching code enables a backpatcher that replaces the corresponding entry in the translated multiple branch-jump table using with pointers to the address of the translated target address, e.g., direct jump backpatch 1010 determines, using any of a variety of know techniques, such as search and compare techniques, whether such ordinal target address has been enter by memory manager 720 in translated instruction look-up table 232 (see Mattson, Abstract, col. 12: 1-51, Look-up table 232, col. 27: 31-42, and related text).

It would have been obvious to a person having ordinary skill in the art at the time of invention was made to use dynamic backpatching determination

through look-up table 232 of Mattson in dynamic instrumentation (e.g., handle to look up address) of Tamches for increasing the efficiencies of the dynamic translation as once accomplished by Mattson (see Mattson, col. 1: 48-63).

As to claim 9, Tamches discloses further comprising: a buffer for storing the data (e.g., L1 data cache see page 84 or kernel space Figure 3.1, or springboard space, Figure 4.9, page 62).

As to claim 10, Tamches discloses further comprising:

a tracing framework configured to emulate the original instruction to determine a value of a program counter if the original instruction is a control-flow instruction and to return control to a thread at an address of the program counter value if the original instruction is a control-flow instruction (e.g., the relocating the overwritten instruction to the code patch semantic section 4.2.1, page 51-52).

As to claim 11, Tamches further discloses wherein the trap handler sets a destination of the jump instruction to a next address immediately following and address of the trap instruction (*e.g., overwrite a single branch or jump instruction x86 by writing a one-byte trap or.... and jump to the appropriate code patch – see page 66-67, section 4.6.1 and code splicing overwrite instruction of Figure 4.9, page 62).*

As to claim 12, Tamches further discloses wherein the scratch space is allocated on a per-thread basis (e.g., event interval accumulation per thread – see page 77 and related text).

As to claim 13, Tamches further discloses wherein the instrumented program is executed on a multi-thread architecture (e.g., even interval accumulation per multiple threads – see page 66 and related text).

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to the applicant disclosure.

Hsu et al. (US 6,295,644 B1) is cited to teach patching program text to improve performance of application.

Pierce et al., "Idtrace – A Tracing Tool for i486 Simulation", University of Michigan, IEEE, 1994, is cited to each Idtrace, a binary instrumentation tool which produces execution traces for the ix86 instruction set architecture.

Bosch et al., "Complete x86 instruction trace generation from hardware bus collect", University of Paris (emphasis added), IEEE, 1997, is cited to teach hardware/software approach to collect perfect x86 traces using a commercial analyzer.

Uhlig, "Trace-Driven Memory Simulation: A survey", University of Michigan, ACM, 1997, is cited to teach surveys and analyzes the trace-driven memory simulation by establishing criteria for evaluating, and then applies the criteria those criteria".

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Marina Lee whose telephone number is (571) 270-1648. The examiner can normally be reached on M-F (11:00 am to 7: 30 pm) Est..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

M.L.
November 2, 2007

TUAN DAM
SUPERVISORY PATENT EXAMINER